LIBRARY MANAGEMENT SYSTEM

ABSTRACT:

With the advancement of technology, it is imperative to exalt all the systems into a user-friendly manner. The Library Management system (LMS) acts as a tool to transform traditional libraries into digital libraries. In traditional libraries, the students/user has to search for books which are hassle process and there is no proper maintenance of database about issues/fines.

The librarians have to work allotted for arranging, sorting books in the book sells. At the same time, they have to check and monitor the lend/borrow book details with its fine. It is a tedious process to work simultaneously in different sectors.

LMS will assist the librarians to work easily. The LMS supports the librarians to encounter all the issues concurrently. The users need not stand in a queue for a long period to return/borrow a book from the library. The single PC contains all the data's in it. The librarians have to assess the system and provide an entry in it. Through LMS the librarian can find the book in the bookshelves.

PROBLEM STATEMENT :

There is a lack of organization in book management, user registration, and book borrowing. The system relies heavily on manual processes, leading to errors in book availability and delays in user requests. Additionally, there is no proper system for tracking overdue books and fines.

The problem faced is that library users require an efficient method to find a specific book or keyword(s) within a book given a continuously expanding library.

BENEFITS :

The benefits of a library management system, particularly Koha, are numerous, such as centralized management, improved access, easy cataloging, efficient circulation, real-time reporting, multi-lingual support, interoperability, community support, and security.

METHODLOGY :

The LMS contains an Admin module where it demonstrates the activates of the admin. Admin is considered as the authorized person to access the LMS system. He/she can access the LMS system through their user id and password. At the time of login, the system is loaded and opens the Home page where he/she has to enter the Id and password .Once he/she login to the system, then they can access/modify the data in it.

KEY FEATURES

1. Membership Administration. ...
2. Scanning & Bar-Coding. ...
3. Online Access. ...
4. Self Management. ...
5. Management Of Fees. ...
6. Catalog Circulation ….
7.

MODULE :

1. Login form Module:
   - User ID and Password to Login into Account

2. Book Cataloging Module :
   - Manages information about books, including title, author, publication date, ISBN, genre, and other relevant details.

3. Payment Module :
   - Manages the payment way whether net banking or UPI or others.,

4. Membership and User Detail Module :

- Manages the details about the user like name, ID, books borrowed, books bought, E-mail address, phone number, gender, DOB.

5. Fines and Fee Management Module :
   - Calculates and manages fines for overdue books.

DATA SET :

1. **Book** :
   - **Books Name** (Book's name)
   - **Books ID** (Unique identifier for each book)
   - **Author Name** (Book's name)
   - **Author ID** (Unique identifier for each Author)
   - **Genre** (Type of Book)
   - **Published Year** (Book published year)
   - **Country or Nationality** (Country name)

2. **User or Buyer:**
   - **User ID** (Unique identifier for each user)
   - **User Name** (User’s name)
   - **User E-mail** (User’s E-mail)
   - **User Phone number** (User’s Phone number)
   - **User Address** (User’s Address)

3. **Rental :**
   - **Books Name** (Book’s name)
   - **Books ID** (Unique identifier for each book)
   - **User Name** (User’s name)
   - **User ID** (Unique identifier for each user)
   - **Buying Date** (Date of buying book)
   - **Return Date** (Date of return book)
   - **Overdue Date** (Number of days exceed after return date)
   - **Cost of Rent** (Final amount for payment)

NEED FOR THE PROJECT :

Database Design:

Design a database schema to store information about books, users, borrowing history, etc. You might have tables for books, users, transactions, etc.

User Interface:

Develop a user interface for interacting with the system. This could be a web application, desktop application, or mobile application depending on your requirements.

Functionality: Implement features such as:

- Adding, updating, and deleting books
- Adding, updating, and deleting users
- Borrowing and returning books
- Searching for books by title, author, category, etc.
- Managing user accounts and permissions
- Generating reports such as overdue books, popular books, etc.

Authentication and Authorization:

Implement a system for user authentication and authorization. Users should be able to log in with their credentials and access only the features they are authorized to use.

Notifications:

Implement a system for sending notifications to users, such as overdue book reminders, reservation notifications, etc.

Integration with External Systems:

Integrate with external systems if needed, such as online book databases for retrieving book information, payment gateways for fines, etc.

Testing:

Perform thorough testing to ensure the system works as expected and is free of bugs. This includes unit testing, integration testing, and user acceptance testing.

Documentation:

Document the system including user manuals, developer documentation, database schema documentation, etc.

Deployment:

Deploy the system to a production environment where users can access it. This might involve setting up servers, databases, security measures, etc.

Maintenance and Support:

Provide ongoing maintenance and support for the system, including bug fixes, updates, and user support.

OBJECTIVE OF THE PROJECT:

Efficient Resource Management:

- Organizing and cataloging library resources such as books, journals, magazines, multimedia materials, etc.

- Maintaining accurate records of available resources, including their titles, authors, editions, categories, and availability status (e.g., checked out, available, reserved).
- Managing inventory and tracking the circulation of resources, including borrowing, renewals, and returns.

Streamlined Operations:

- Automating routine library operations such as check-in, check-out, reservations, and renewals to reduce manual effort and errors.
- Providing self-service options for library users to search for resources, place holds, renew items, and pay fines.
- Generating reports and statistics to monitor library usage, track trends, and make informed decisions about collection development and resource allocation.

Enhanced User Experience:

- Providing a user-friendly interface for library patrons to easily search and access library resources, check availability, and manage their accounts.
- Offering personalized features such as recommendations based on past borrowing history or interests, alerts for overdue items, and notifications for reserved items becoming available.
- Supporting multiple access points, including web-based interfaces, mobile applications, and self-service kiosks, to accommodate diverse user preferences and needs.

Effective Information Retrieval:

- Implementing advanced search capabilities, including keyword search, title search, author search, subject search, and advanced filtering options to facilitate efficient information retrieval.
- Integrating with external databases or APIs to enhance the library catalog with additional metadata, cover images, reviews, and other relevant information.

Administrative Oversight:

- Providing administrative tools for library staff to manage user accounts, configure system settings, generate reports, and perform other administrative tasks.
- Supporting role-based access control to ensure that staff members have appropriate permissions and access levels based on their roles and responsibilities.

Scalability and Adaptability:

- Designing the system to scale gracefully as the library collection grows and user demands evolve over time.
- Building a modular and flexible architecture that allows for easy customization, integration with external systems, and future enhancements.

SCOPE OF THE PROJECT

Resource Management:

- Cataloging and organizing library resources such as books, journals, magazines, multimedia materials, etc.
- Maintaining detailed records of each resource, including titles, authors, editions, categories, and availability status.
- Tracking the circulation of resources, including borrowing, renewals, reservations, and returns.

User Interactions:

- Providing a user-friendly interface for library patrons to search for resources, check availability, and manage their accounts.
- Allowing users to place holds, renew items, pay fines, and receive notifications about overdue items or reserved items becoming available.
- Supporting self-service options such as self-checkout kiosks and online account management portals.

Administrative Functions:

- Offering administrative tools for library staff to manage user accounts, configure system settings, generate reports, and perform other administrative tasks.
- Implementing role-based access control to ensure that staff members have appropriate permissions and access levels based on their roles and responsibilities.
- Providing robust reporting capabilities to monitor library usage, track trends, and make data-driven decisions about collection development and resource allocation.

Integration and Customization:

- Integrating with external databases, APIs, or library consortia to enhance the library catalog with additional metadata, cover images, reviews, and other relevant information.
- Supporting customization options to tailor the system to the specific needs and preferences of the library and its users.
- Ensuring compatibility with existing library systems, protocols, and standards to facilitate seamless integration and interoperability.

Accessibility and Scalability:

- Designing the system to be accessible to users with diverse needs, including those with disabilities or language preferences.
- Building a scalable architecture that can accommodate growth in the library collection, user base, and system usage over time.
- Ensuring high availability and reliability to minimize downtime and disruptions to library services.

Security and Privacy:

- Implementing robust security measures to protect sensitive user data, including personal information, borrowing history, and financial transactions.
- Complying with relevant regulations and standards for data privacy and security, such as GDPR, HIPAA, and PCI DSS.

User Experience Enhancement:

- Continuously improving the user experience through user feedback, usability testing, and iterative design enhancements.
- Incorporating features such as personalized recommendations, advanced search capabilities, and seamless navigation to enhance user satisfaction and engagement.

LITERATURE SURVEY:

A literature survey on library management encompasses an exploration of existing research and scholarly works focusing on the various facets of library operations. This involves delving into studies that investigate the adoption and integration of technology within libraries, including the implementation of Library Management Systems (LMS) and digital tools to streamline processes and enhance user experiences. Additionally, the survey entails a review of best practices in collection development, cataloging, circulation, and user services, along with an analysis of innovative strategies employed by libraries to meet evolving user needs and expectations.

Moreover, the literature survey aims to identify common challenges encountered by libraries in managing their resources and services effectively. This may include issues related to budget constraints, space limitations, and the digital transformation of library collections. By examining literature on the solutions and strategies employed to address these challenges, researchers can gain valuable insights into practical approaches for optimizing library operations and improving service delivery. Furthermore, the survey explores emerging trends and future directions in library management, such as the utilization of data analytics, artificial intelligence, and virtual reality, providing valuable insights into the evolving landscape of library services and resources.

3.SYSTEM DESIGN :

The system design for a Library Management System (LMS) involves outlining the architecture, modules, and functionality of the system to effectively manage library resources and operations. Here's a brief overview of the system design

3.1 PROPOSED SYSTEM ARCHITECTURE :

The LMS typically follows a client-server architecture, where the server hosts the database and core functionality, while clients (such as web browsers or mobile apps) interact with the system to perform various tasks. The architecture should be scalable and flexible to accommodate future growth and changes in requirements.

3.2 Modules and Functionality:

User Management: This module handles user authentication, registration, and account management. It allows patrons to create accounts, update their profiles, and access library services.

User Management Module:

This module handles user authentication, registration, and account management.

Functionalities include user registration, login, password management, profile updates, and account deletion.

It may also include features for managing user roles and permissions, such as assigning different access levels for patrons and library staff.

Cataloging and Inventory Management Module:

This module is responsible for managing library resources and maintaining a catalog of available items.

Functionalities include adding new items to the catalog, updating item details (such as title, author, category, etc.), and deleting or archiving items.

It may also include features for managing copies of items, tracking item availability, and handling inter-library loans.

Circulation and Borrowing Module:

This module facilitates the borrowing and lending of library materials.

Functionalities include searching for items, placing holds or reservations, checking out items, renewing loans, and returning items.

It may also include features for managing fines and fees, handling overdue items, and sending notifications to users about upcoming due dates.

Administration Module:

This module provides administrative tools for managing system settings and configurations.

Functionalities include managing user accounts, configuring library policies (such as loan periods and borrowing limits), and generating reports.

It may also include features for managing library branches, staff accounts, and circulation rules.

Search and Discovery Module:

This module enables users to search for and discover library resources based on various criteria.

Functionalities include keyword search, advanced search options (such as filtering by author, subject, or publication date), and sorting search results.

It may also include features for browsing categories, viewing recommended items, and saving searches for future reference.

Notifications and Alerts Module:

This module sends notifications and alerts to users about important events or updates.

Functionalities include sending notifications about overdue items, reserved items becoming available, system downtime, or policy changes.

It may also include features for managing communication preferences and scheduling automated reminders.

Reporting and Analytics Module:

This module generates reports and analytics to monitor library usage, track trends, and make data-driven decisions.

Functionalities include generating circulation reports, analyzing user demographics, and evaluating collection usage.

It may also include features for exporting data, visualizing trends through charts or graphs, and identifying areas for improvement.

4.REQUIREMENT SPECIFICATION :

4.1 HARDWARE REQUIREMENT

Hardware requirements for a library management system (LMS) depend on factors such as the scale of the library, the number of users, and the functionalities supported by the system. Here's a general outline of the hardware requirements for an LMS:

Server Infrastructure:

Central Processing Unit (CPU): A multi-core processor capable of handling concurrent user requests and database operations efficiently.

Random Access Memory (RAM): Sufficient RAM to support the operating system, database management system (DBMS), and caching mechanisms. The amount of RAM required depends on the size of the library's database and the expected workload.

Storage: Adequate storage space for storing the LMS application files, database files, and backups. The storage should be scalable to accommodate future growth in data volume.

Network Interface: Fast and reliable network connectivity to ensure seamless communication between server components and client devices.

Database Server:

Depending on the size of the library's database and the expected number of concurrent users, the database server should have sufficient CPU, RAM, and storage resources to handle database queries, transactions, and data retrieval operations efficiently.

It's recommended to use a high-performance, reliable database management system (DBMS) such as MySQL, PostgreSQL, or Oracle to store and manage library data securely.

Client Devices:

Patron Workstations: Desktop computers, laptops, or mobile devices used by library patrons to access the LMS interface, search for library resources, and perform various library-related tasks.

Library Staff Workstations: Desktop computers or laptops used by library staff to manage the LMS, perform administrative tasks, and provide assistance to patrons.

Barcode Scanners: If the library uses barcode technology for inventory management and circulation tasks, barcode scanners may be required at circulation desks and self-checkout stations.

Network Infrastructure:

Local Area Network (LAN): A robust LAN infrastructure to connect client devices to the server and facilitate communication between different components of the LMS.

Internet Connectivity: Stable and high-speed internet connectivity to enable remote access to the LMS, online catalog search, and access to electronic resources and databases.

Backup and Redundancy:

Backup Server: A dedicated backup server or storage device to perform regular backups of the LMS database and application files.

Redundant Components: Redundant power supplies, RAID configurations, and failover mechanisms to ensure high availability and data integrity in case of hardware failures or system crashes.

4.2 SOFTWARE REQUIREMENT

Software requirements for a library management system (LMS) encompass the necessary software components, platforms, and technologies needed to develop, deploy, and maintain the system. Here's an overview of the software requirements for an LMS:

Operating System:

Server Operating System: The server-side components of the LMS typically run on a server-grade operating system such as:

- Linux (e.g., Ubuntu Server, CentOS)
- Windows Server
- Unix (e.g., FreeBSD)

Client Operating System: The client-side components of the LMS should be compatible with various operating systems, including:

- Windows
- macOS
- Linux
- Mobile Operating Systems (iOS, Android)

Web Server:

The LMS may utilize a web server to host the web-based interface accessible to patrons and library staff. Commonly used web servers include:

- Apache HTTP Server
- Nginx
- Microsoft Internet Information Services (IIS)

Database Management System (DBMS):

The LMS requires a robust DBMS to store and manage library data efficiently. Commonly used database systems for LMS include:

- MySQL
- PostgreSQL
- Microsoft SQL Server
- Oracle Database
- SQLite (for smaller-scale deployments)

Programming Languages and Frameworks:

Back-End Development: Programming languages and frameworks are used to develop the server-side logic and backend components of the LMS. Common choices include:

- Java (e.g., Spring framework)
- Python (e.g., Django framework)
- PHP (e.g., Laravel framework)
- Node.js (e.g., Express.js framework)

Front-End Development: Technologies for building the client-side interface and user experience include:

- HTML5
- CSS3
- JavaScript (e.g., React.js, Angular, Vue.js)

Integrated Development Environment (IDE):

Developers may use IDEs or text editors to write, debug, and manage the codebase of the LMS. Popular IDEs for various programming languages include:

- IntelliJ IDEA (for Java)
- PyCharm (for Python)
- Visual Studio Code
- Eclipse
- NetBeans

Version Control System:

Version control systems are used to track changes to the source code, collaborate with team members, and manage project versions. Common version control systems include:

- Git (with platforms like GitHub, GitLab, Bitbucket)
- Subversion (SVN)
- Mercurial

Additional Software Components:

Depending on specific requirements, the LMS may integrate additional software components such as:

- Barcode Scanning Software

- PDF Generation Libraries
- Authentication and Authorization Libraries (e.g., OAuth, LDAP)
- Reporting and Analytics Tools
- Email Sending Libraries or Services

5. IMPLEMENTATION:

5.1 SAMPLE CODE

FRONT END : LOGINN PAGE

```
<!DOCTYPE html>
<html lang="en">
{% load static %}
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Library</title>
    <link rel="stylesheet" href="{% static 'style1.css' %}">
    <link rel="icon" type="imag/png" href="{% static 'iconic.png' %}">
</head>
<body>
    <header>
        <h1>LIBRARY</h1>
        <nav>
            <li><a href="home.html">Home</a></li>
            <li><a href="about.html">About us</a></li>
            <li><a href="contact.html">To Contact us</a></li>
        </nav>
    </header>
    <img src="{% static 'login.jpg' %}" alt="photo" id="ph1">
    <div id="a1">
        <h3>Login Page</h3>
        <style>
            h4{
```

```
        font-size: 20px;
        text-align: center;
        color: rgba(213, 172, 67);
        margin-top: -20px;
        margin-bottom: -1px;
        }
    </style>
    {% for message in messages %}
        <h4>
            {{ message }}
        </h4>
    {% endfor %}
    <form method="POST" action="login">
        {% csrf_token %}
        <div id="username">
            <input id="un" type="text" placeholder="enter username">
        </div>
        <div id="pass">
            <input id="ps" type="password" minlength=8 placeholder="enter password">
        </div>
        <div id="userid">
            <input id="uid" type="text" placeholder="enter id">
        </div>
        <div id="userid">
            <input id="ueid" type="email" placeholder="Enter your Email-Id">
        </div>
        <button type="submit" id="sub">LOGIN</button>
        <a href="register.html"><button type="submit" id="reg">REGISTER</button></a>
    </form>
</div>
```

```
</body>
</html>
```

REGISTERATION PAGE :

```
<!DOCTYPE html>
<html lang="en">
{% load static %}
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Library</title>
    <link rel="stylesheet" href="{% static 'style1.css' %}">
    <link rel="icon" type="imag/png" href="{% static 'iconic.png' %}">
</head>
<body>
    <header>
        <h1>LIBRARY</h1>
        <nav>
            <li><a href="home.html">Home</a></li>
            <li><a href="about.html">About us</a></li>
            <li><a href="contact.html">To Contact us</a></li>
            <li><a href="contact.html">My Profile</a></li>
        </nav>
    </header>
    <img src="{% static 'login.jpg'  %}" alt="photo" id="ph1">
    <div id="a1">
        <h3>REGISTRATION PAGE</h3>
        <style>
            h4{
            font-size: 20px;
            text-align: center;
```

```
            color: rgba(213, 172, 67);
            margin-top: -20px;
            margin-bottom: -1px;
            }
        </style>
        {% for message in messages %}
            <h4>
                {{ message }}
            </h4>
        {% endfor %}
        <form method="POST" action="register">
            {% csrf_token %}
            <div id="username">
                <input type="text" name="username" placeholder="enter username" />
            </div>
            <div id="pass">
                <input type="password" name="password1" minlength=8 placeholder="enter
password"/>
            </div>
            <div id="pass1">
                <input type="password" name="password2" placeholder="confirm the
password"/>
            </div>
            <div id="userid">
                <input type="email" name="email" placeholder="Enter your Email-Id"/>
            </div>
            <button type="submit" id="reg1">REGISTER</button >
        </form>
    </div>
</body>
</html>
```

BOOKS AVAILABLE :

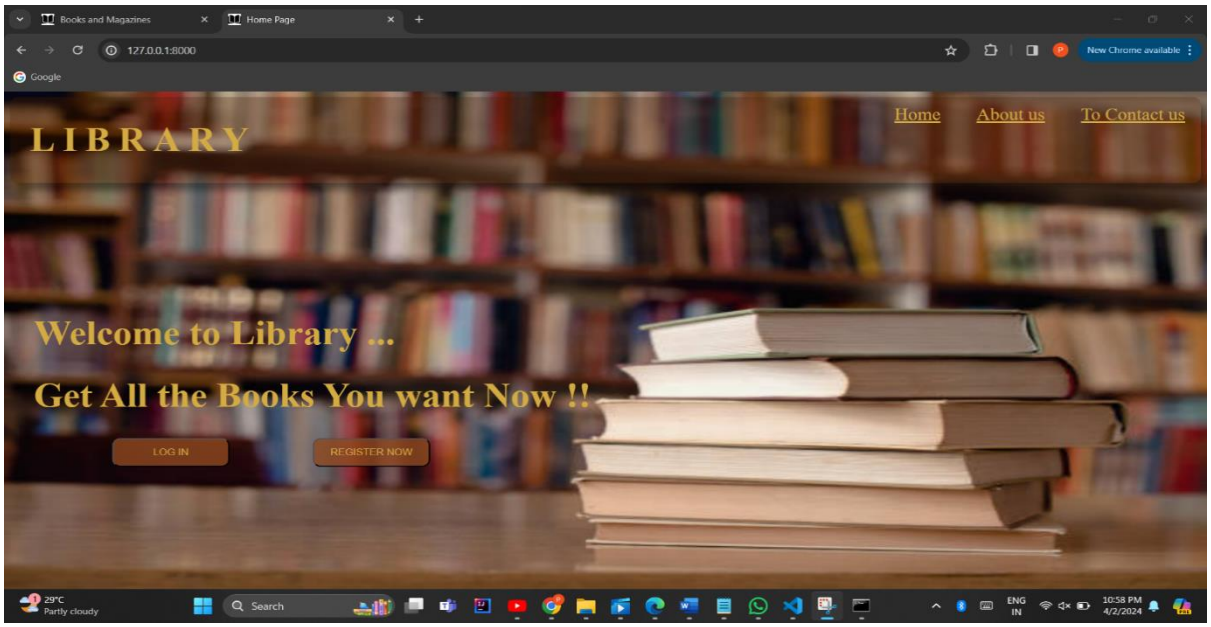
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Books and Magazines </title>
    <link rel="stylesheet" href="style1.css">
    <link rel="icon" type="imag/png" href="iconic.png">
</head>
<body>
    <script src="js1.js"></script>
    <header>
        <h1><a href="index.html">LIBRARY</a></h1>
        <nav>
            <li><a href="home.html">Home</a></li>
            <li><a href="about.html">About us</a></li>
            <li><a href="contact.html">To Contact us</a></li>
            <li><a href="Profile.html" onclick="n2()">My Profile</a></li>
        </nav>
    </header>
    <img src="login.jpg" id="ph1">
    <div  class="cont">
        <div class="containers">
            <div>
                <a href="desc1.html"><img src="wings.jpg" alt="wings of fire" id="pics"
title="Wings of fire"></a>
            </div>
            <div id="contents">
                <h1>Wings Of Fire</h1>
                <h3><ul>Writter : Dr.A.P.J Abdul Kalam</ul></h3>
```

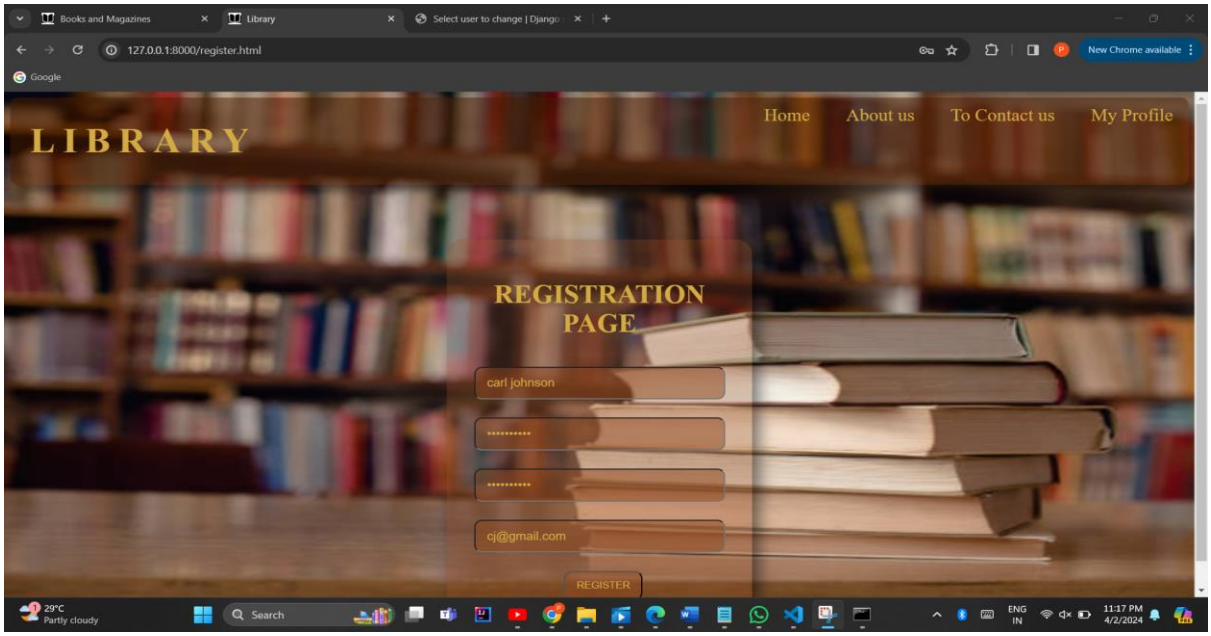
```
            <h3><ul>Published Year : 1999</ul></h3>
            <h3><ul>Genre : Autobiography</ul></h3>
        </div>
    </div>
    <div class="containers">
        <div>
            <a href="desc2.html"><img src="discoveryofindia.jpg" alt="wings of fire"
id="pics" title="The Discovery of India"></a>
        </div>
        <div id="contents">
            <h1>The Discovery of India</h1>
            <h3><ul>Writter : Jawaharlal Nehru</ul></h3>
            <h3><ul>Published year : 1946</ul></h3>
            <h3><ul>Genre : history and culture.</ul></h3>
        </div>
    </div>
    <div class="containers">
        <div>
            <a href="desc3.html"><img src="ponni.jpeg" alt="Ponniyan Selven" id="pics"
title="Ponniyin Selvan"></a>
        </div>
        <div id="contents">
            <h1>Ponniyin Selvan</h1>
            <h3><ul>Writter : Kalki Krishnamurthy. </ul></h3>
            <h3><ul>Published Year : 1955.</ul></h3>
            <h3><ul>Genre : historical fiction.</ul></h3>
        </div>
    </div>
    <div class="containers">
        <div>
```

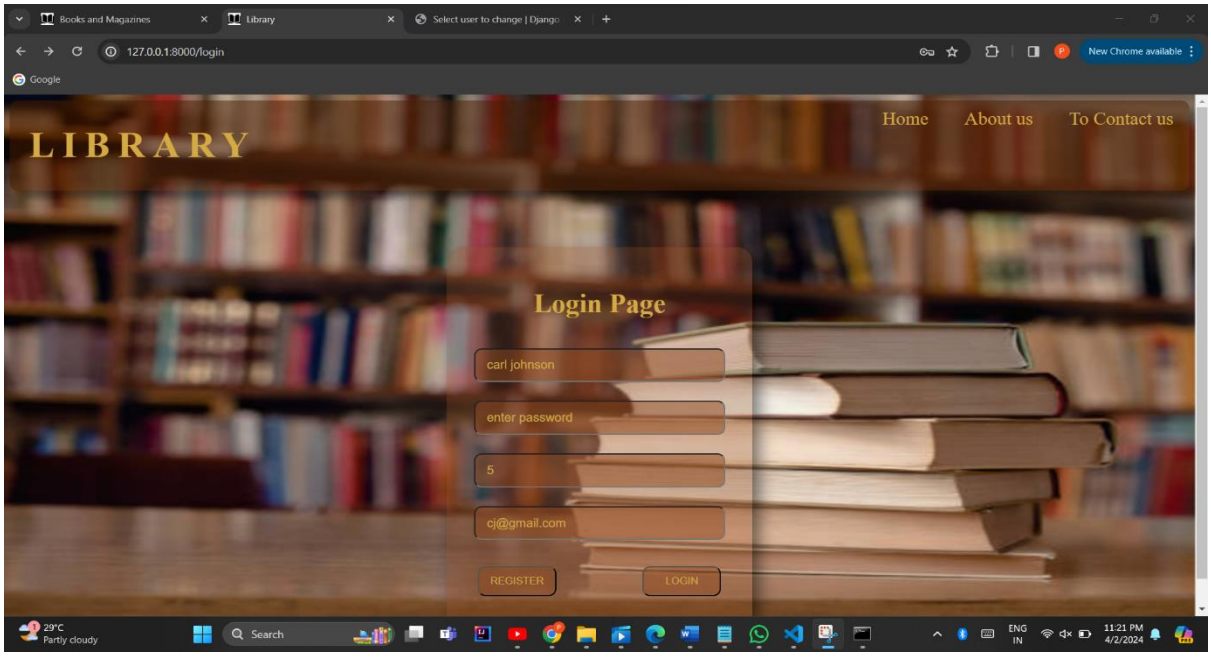
```
            <a href="desc4.html"><img src="tokillamok.jpg" alt="" id="pics" title="To Kill a Mockingbird"></a>
        </div>
        <div id="contents">
            <h1>To Kill a Mockingbird</h1>
            <h3><ul>Writter : Harper Lee</ul></h3>
            <h3><ul>Published : 1960.</ul></h3>
            <h3><ul>Genre : Classic Multigenre Novel</ul></h3>
        </div>
    </div>
    <div class="containers">
        <div>
            <a href="desc5.html"><img src="wolfhall.jpeg" alt="" id="pics" title="Wolf Hall"></a>
        </div>
        <div id="contents">
            <h1>Wolf Hall</h1>
            <h3><ul>Written : Hilary Mantel</ul></h3>
            <h3><ul>Published : 2009</ul></h3>
            <h3><ul>Genre : historical fiction.</ul></h3>
        </div>
    </div>
    <div class="containers">
        <div>
            <a href="desc6.html"><img src="hsb.jpeg" alt="" id="pics" title="Heart-Shaped Box"></a>
        </div>
        <div id="contents">
            <h1>Heart-Shaped Box</h1>
            <h3><ul>Writter : Joe Hill</ul></h3>
            <h3><ul>Published : 2007</ul></h3>
```

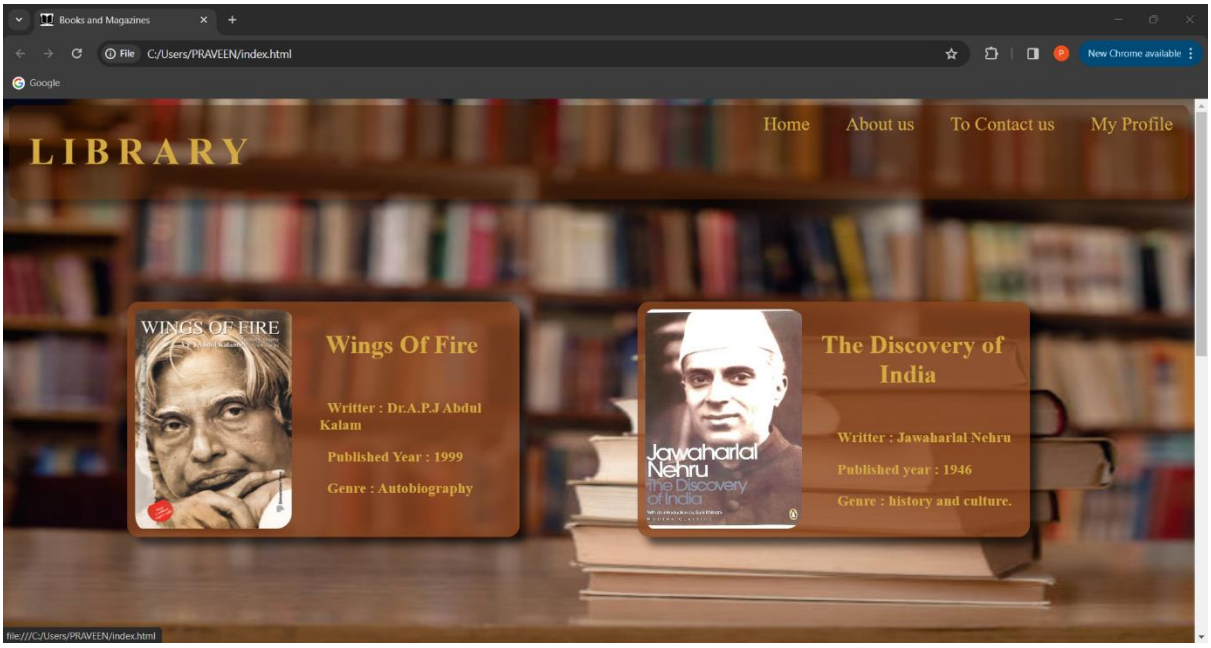
```
            <h3><ul>Genre : Horror Novel</ul></h3>
          </div>
        </div>
    </div>
</body>
</html>
```

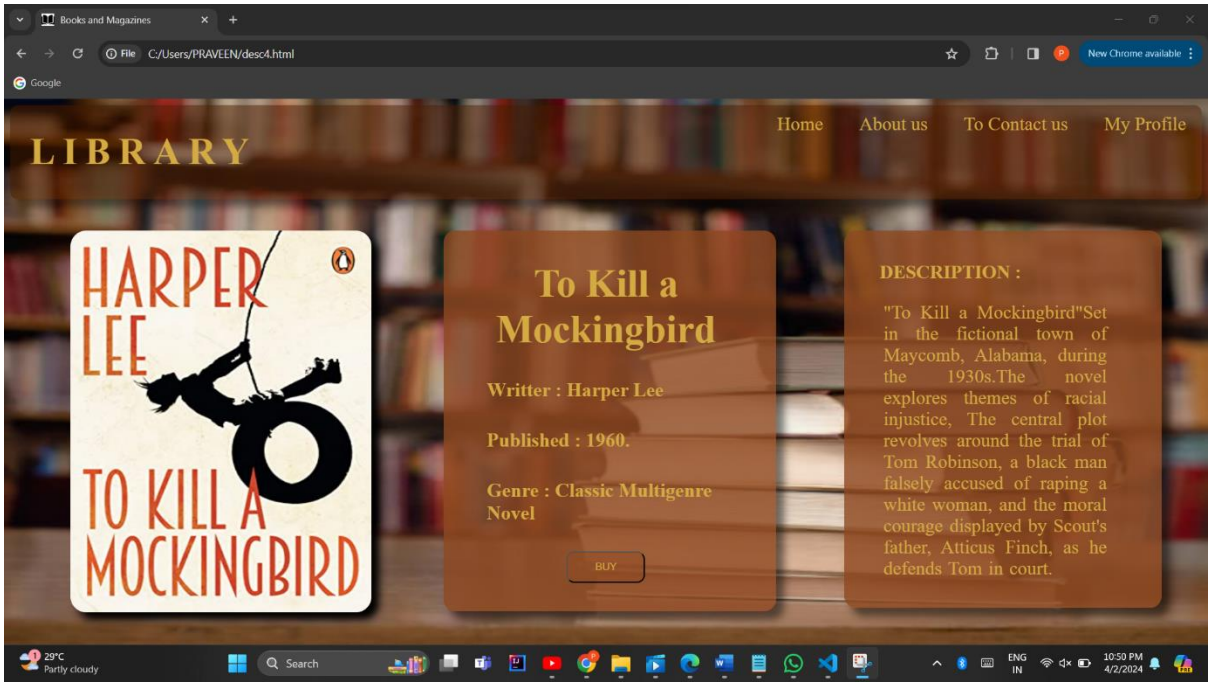
5.2 SAMPLE SCREEN SHOTS :

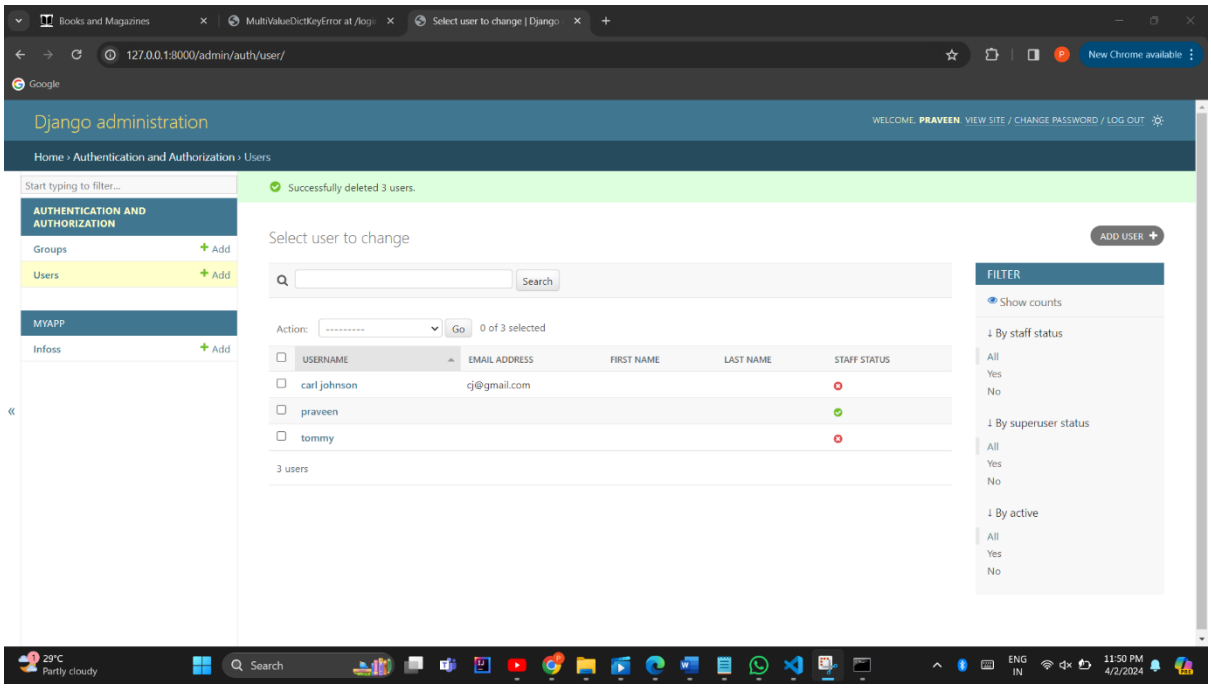


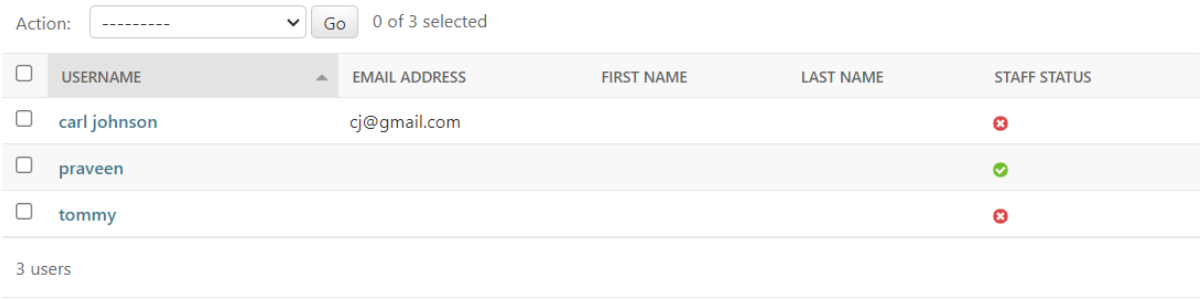



Books and Magazines
Library
Select user to change | Django
127.0.0.1:8000/login
New Chrome available
Google
LIBRARY
Home
About us
To Contact us
Login Page
carl johnson
enter password
5
cj@gmail.com
REGISTER
LOGIN
29°C
Partly cloudy
Search
ENG
IN
11:21 PM
4/2/2024


Books and Magazines
File
C:/Users/PRAVEEN/index.html
New Chrome available
Google
LIBRARY
Home
About us
To Contact us
My Profile
WINGS OF FIRE
Wings Of Fire
Writter : Dr.A.P.J Abdul Kalam
Published Year : 1999
Genre : Autobiography
Jawaharlal Nehru
The Discovery of India
The Discovery of India
Writter : Jawaharlal Nehru
Published year : 1946
Genre : history and culture.
file:///C:/Users/PRAVEEN/index.html


Books and Magazines
File
C:/Users/PRAVEEN/desc4.html
New Chrome available
Google
LIBRARY
Home
About us
To Contact us
My Profile
HARPER LEE
TO KILL A MOCKINGBIRD
To Kill a Mockingbird
Writter : Harper Lee
Published : 1960.
Genre : Classic Multigenre Novel
BUY
DESCRIPTION :
"To Kill a Mockingbird"Set in the fictional town of Maycomb, Alabama, during the 1930s.The novel explores themes of racial injustice, The central plot revolves around the trial of Tom Robinson, a black man falsely accused of raping a white woman, and the moral courage displayed by Scout's father, Atticus Finch, as he defends Tom in court.
29°C
Partly cloudy
Search
ENG
IN
10:50 PM
4/2/2024





Panimalar Engineering College, Chennai.		
Department of Information Technology		
Title	Max. Marks	Marks Awarded
Quality of Work / Performance	4	
Viva voce	2	
Record	4	
Total	10	
Submitted Date		
Staff Signature		

Result:

The implementation of application for the online course registeration system was successfully developed and executed